

## Fetch: Cross-Origin Requests

- From the browser point of view, there are two kinds of cross-origin requests: "safe" and all the others.
- "Safe" requests must satisfy the following conditions:
- Method: GET, POST or HEAD.
- Headers – we can set only:
- Accept
- Accept-Language
- Content-Language
- Content-Type to the value application/x-www-form-urlencoded, multipart/form-data or text/plain.
- The essential difference is that safe requests were doable since ancient times using `<form>` or `<script>` tags, while unsafe were impossible for browsers for a long time.
- So, the practical difference is that safe requests are sent right away, with the Origin header, while for the other ones the browser makes a preliminary "preflight" request, asking for permission.
- **For safe requests:**
- → The browser sends the Origin header with the origin.
- ← For requests without credentials (not sent by default), the server should set:
- Access-Control-Allow-Origin to \* or same value as Origin
- ← For requests with credentials, the server should set:
- Access-Control-Allow-Origin to same value as Origin
- Access-Control-Allow-Credentials to true
- 

## PARTTEN AND FLAGS

- A regular expression consists of a pattern and optional flags: g, i, m, u, s, y.
- Without flags and special symbols (that we'll study later), the search by a regexp is the same as a substring search.
- The method `str.match(regexp)` looks for matches: all of them if there's g flag, otherwise, only the first one.
- The method `str.replace(regexp, replacement)` replaces matches found using regexp with replacement: all of them if there's g flag, otherwise only the first one.
- The method `regexp.test(str)` returns true if there's at least one match, otherwise, it returns false.

